IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Xiaofan LIN et al.	§	Confirmation No.:	8539
Serial No.:	10/774,321	§	Group Art Unit:	3628
Filed:	02/06/2004	§	Examiner:	Fadey S. Jabr
For:	Merit-Based Software Licensing	§	Docket No.:	200310312-1

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: April 8, 2008

Sir/Madam:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed on April 2, 2008.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas. HPDC is a wholly owned affiliate of Hewlett-Packard Company (HPC). The Assignment from the inventors to HPC was recorded on February 6, 2004, at Reel/Frame 014974/0079.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Originally filed claims:	1-27.
Canceled claims:	None.
Presently rejected claims:	1-27.
Presently appealed claims:	1-27.

IV. STATUS OF AMENDMENTS

There have been no amendments following the Final Office Action dated January 2, 2008.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Various embodiments of the invention are described below. The scope of disclosure is not limited by the descriptions of the embodiments that follow. Citations to the specification have been provided to demonstrate where support may be found in the specification for various parts of the invention. Additional support may be found elsewhere in the application.

Appellants' contribution is directed to a technique for determining software licensing fees based on the quality of the software to be licensed. In general, the technique includes determining the performance quality of the software (*e.g.*, by comparing its performance to that of comparable software, by determining an absolute value of performance quality, and using various other techniques) and using the performance quality to calculate a suitable licensing fee.

Claim 1 is directed to a processor-based method that comprises determining a quality value 150 for a target software 120 based on performance of the target software 120 (Fig. 1; p. 3, l. 3 – p. 6, l. 23; p. 7, ll. 13-22) and computing a merit-based licensing fee 170 for the target software 120 based on the quality value 150. Fig. 1; p. 6, l. 24 – p. 7, l. 12; p. 7, ll. 22-24.

Claim 16 is directed to a system 200 that comprises a CPU 210 and a storage device 220 coupled to the CPU 210 and containing executable code 230. Fig. 2; p. 7, ll. 13-17. Upon executing the code 230, the CPU 210 computes a merit-based licensing fee 170 for a target software 120 based on a quality value 150 associated with the target software 120. Figs. 1-2; p. 3, ll. 3-19; p. 7, ll. 20-24.

Claim 22 is directed to a storage device 220 containing software 230 that, when executed by a processor 210, causes the processor 210 to store operation logs 240 of a target software 120 (Figs. 1-2; p. 4, ll. 3-7; p. 7, ll. 17-20), measure a performance level of the target software 120 based on the operation logs 240 (Figs. 1-2; p. 4, ll. 7-10), determine a quality value 150 for the target software 120 based on the performance level (Fig. 1; p. 3, ll. 6-14; p. 3, l. 20 – p. 4, l. 2; p. 4, l. 26 – p. 6, l. 23), and compute a licensing fee 170 for the target software 120 based on the quality value 150 (Fig. 1; p. 3, ll. 14-19; p. 7, l. 22-24).

Claim 25 is directed to a system 200 that comprises means 140, 210, 230, 240, 250 for measuring a performance level of a target software 120 (Figs. 1-2; p. 4, ll. 7-10; p. 7, ll. 13-20), means 210, 230, 240, 250 for determining a quality value 150 for the target software 120 based on the performance level (Figs. 1-2; p. 3, ll. 6-14; p. 3, l. 20 – p. 4, l. 2; p. 4, l. 26 – p. 6, l. 23; p. 7, ll. 20-22), and means 210, 230, 240, 250 for computing a licensing fee 170 for the target software 120 based on the quality value 150 (Figs. 1-2; p. 3, ll. 14-19; p. 7, l. 22-24).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether under 35 U.S.C. § 103(a) claims 1-2, 4-8, 10-11, 13, 15-19 and 21-27 are rendered obvious by Shuster (U.S. Pub. No. 2005/0097059) in view of Miller (U.S. Pub. No. 2004/0261070).

Whether under 35 U.S.C. § 103(a) claims 3, 9, 12, 14 and 20 are rendered obvious by Shuster in view of Miller and Official Notice.

VII. ARGUMENT

A. Summary of Shuster

Shuster is directed to a system by which consumers of digital music files (*e.g.*, MP3 files) may purchase one or more licenses for digital music files that have been downloaded without a proper license. *See* Shuster, paragraph [0013]. Shuster teaches that the price for a license is determined by analysis of the digital music file. In particular, the digital music file may be analyzed based on a sampling rate of the music file, length of the file, version of the file and the type of work downloaded. Paragraph [0037]. Clearly, the type of analysis taught by Shuster involves analyzing the digital music file itself. Metrics (*e.g.*, sampling rate, length of file, version of file) of the digital music file that are electronic properties of the file are used to determine the price for a license. This type of digital file analysis is part of the principle of operation of Shuster, because Shuster uses these analyses in generating checksums of the digital music file, which are later used to calculate a licensing price. Paragraphs [0032]-[0035]. Shuster does not teach or even suggest actually **performing** (*e.g.*, playing) the digital music file to measure metrics or to determine the price for a license. Performance of the file is not needed and thus is not taught by Shuster.

B. Summary of Miller

Miller requires the actual performance of software to detect software bugs. Miller is directed to a system that automatically detects and corrects software bugs. Miller clearly teaches that the detection of software bugs is done by a monitoring system that "watches" while multiple persons use the software. In particular, "as the software . . . is being used, its performance is automatically monitored based on predetermined monitoring criteria." Miller, paragraph [0007]. Unlike Shuster, whose principle of operation includes non-performance analysis of digital music files, Miller teaches that the actual **performance** of software is used to determine the quality of the software.

C. Rejections under 35 U.S.C. § 103(a)

1. The Examiner Erred in Rejecting Claims 1-2, 4-8, 10-11, 13, 15-19 and 21-27 under Shuster in View of Miller Because Combining Shuster and Miller Impermissibly Alters Shuster's Principle of Operation and Renders Shuster Inoperable for its Intended Purpose

a) General Remarks

The Examiner rejected claims 1-2, 4-8, 10-11, 13, 15-19 and 21-27 as allegedly obvious under Shuster in view of Miller. Independent claim 1 is representative of this group of claims. The grouping should not be construed to mean the patentability of any of the claims may be determined in later actions (*e.g.*, actions before a court) based on the groupings. Rather, the presumption of 35 USC § 282 shall apply to each of these claims individually.

Claim 1 requires "determining a quality value for a target software based on performance of the target software." Claim 1 further requires "computing a merit-based licensing fee for the target software based on the quality value." The Examiner admits that Shuster fails to disclose "determining a quality value for the target software based on the performance level" and so the Examiner turns to Miller. However, Shuster and Miller cannot be combined for at least two reasons: **first**, modifying Shuster with Miller impermissibly alters Shuster's principle of operation, and **second**, modifying Shuster with Miller renders Shuster inoperable for its intended purpose. Each of these reasons is now addressed in turn.

According to MPEP § 2143.01 (VI), a 35 U.S.C. § 103(a) rejection is improper if it modifies a reference by changing the reference's principle of operation: "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." Adding Miller to Shuster impermissibly alters Shuster's principle of operation; thus, Shuster and Miller cannot be used to reject claim 1.

As Appellants explained above, and now reiterate, Shuster is directed to a system by which consumers of digital music files (*e.g.*, MP3 files) may purchase

licenses for digital music files that have been downloaded without a proper license. *See* Shuster, paragraph [0013]. The price for a license is determined by analysis of the digital music file. The digital music file is analyzed based on a sampling rate of the music file, length of the file, version of the file and the type of work downloaded. Paragraph [0037]. The type of analysis taught by Shuster involves analyzing the digital music file itself. Metrics (*e.g.*, sampling rate, length of file, version of file) of the digital music file that are electronic properties of the file are used to determine the price for a license. **This type of digital file analysis is part of Shuster's principle of operation, because Shuster uses these analyses in generating checksums of the digital music file, which are later used to calculate a licensing price**. Paragraphs [0032]-[0035]. Shuster does not teach or even suggest actually **performing** (*e.g.*, playing) the digital music file to measure metrics or to determine the price for a license. Performance of the file is not needed and thus is not taught by Shuster.

In fact, performance of a digital music file in Shuster would be nonsensical. For example, performance of the digital music file might entail a computer playing the file over speakers and then capturing the audible music via a microphone for analysis. Such a technique would be inefficient and unnecessary in Shuster's system. In another example, performance of the digital music file in Shuster might entail playing the music file to a group of human listeners who then rate the music file quality and provide their ratings to the computer that plays the music. However, again, such a technique would be inefficient and unnecessary for Shuster.

By contrast, Miller requires the actual performance of software to automatically detect and correct software bugs. Miller clearly teaches that the detection of software bugs is done by a monitoring system that "watches" while multiple persons use the software. In particular, "as the software . . . is being used, its performance is automatically monitored based on predetermined monitoring criteria." Miller, paragraph [0007].

Unlike Shuster, whose principle of operation includes the **non-performance** analysis of digital music files, Miller teaches that the actual **performance** of software is used to determine the quality of the software. Thus, combining Miller, which teaches performance-based analysis, with Shuster, which teaches non-performance-based analysis, would impermissibly alter Shuster's principle of operation in violation of MPEP § 2143.01(VI).

Further, the combination of Miller and Shuster renders Shuster inoperable for its intended purpose in violation of MPEP § 2143.01(V), which states that "[i]f proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification." As explained above, Shuster's intended purpose is the **automatic analysis** of digital music files to determine licensing costs. Miller, however, teaches that software is analyzed via the monitoring of actual software performance. Therefore, combining Miller's method of analysis with Shuster renders Shuster inoperable for its intended purpose, in violation of MPEP § 2143.01(V).

Based on the foregoing, Appellants respectfully submit that the Examiner erred in rejecting claim 1. Because claim 1 is patentable over the combination of Shuster and Miller, Appellants respectfully submit that all claims in the grouping of claim 1 also are patentable over the combination of Shuster and Miller.

b) Responses to Examiner's Specific Arguments

The Examiner argues that because Shuster teaches "that other defects in the copy, such as background hiss indicating that the data has once been stored in analog, or encoding defects such as pops may also influence the price calculation," Shuster teaches determining a quality value of the target software "based on performance of the target software," as required by claim 1. The Examiner assumes that the analysis of background hiss and encoding defects (*e.g.*, pops) constitutes the "performance" required by claim 1. However, Appellants point out that the analysis of background hiss and encoding defects does not necessarily mean that the software is actually "performed." As is well

known, digital music files can be analyzed using any of a variety of software which graphically illustrate background hiss and encoding defects based on the arrangement of binary bits encoded onto the file. Background hiss and encoding defects may be represented by stray bits that can be detected and analyzed using a computer without ever actually performing the software. Thus, actual performance is not taught or suggested. Absent any explicit, inherent or even implicit teaching or suggestion that the software is actually "performed," Appellants maintain that all pending claims are patentable over the combination of Shuster and Miller.

The Examiner also refers to *In re Keller* and argues that

> [t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art.

The Examiner appears to be quoting MPEP § 2145(III). However, Appellants note with interest that the Examiner neglected to include the final sentence of this MPEP section, which states:

> However, the claimed combination cannot change the principle of operation of the primary reference or render the reference inoperable for its intended purpose.

Appellants have already established that the Examiner's combination changes Shuster's principle of operation and renders Shuster inoperable for its intended purpose. Thus, the Examiner's argument is moot.

2. The Examiner Erred in Rejecting Claims 3, 9, 12, 14 and 20 under Shuster in View of Miller and Official Notice Because Combining Shuster and Miller Impermissibly Alters Shuster's Principle of Operation and Renders Shuster Inoperable for its Intended Purpose

Claims 3, 9, 12 and 14 depend on claim 1 and claim 20 depends on claim 16. As explained above, claims 1 and 16 are patentable over the combination of

Shuster and Miller. Thus, the Examiner erred in rejecting claims 3, 9, 12, 14 and 20 using Shuster and Miller.

D. Conclusion

For the reasons stated above, Appellants respectfully request that the rejections be reversed, and the claims set for issuance. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/Nick P. Patel/

Nick P. Patel
PTO Reg. No. 57,365
Conley Rose, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
AGENT FOR APPELLANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

VIII. CLAIMS APPENDIX

1. (Original) A processor-based method, comprising:

determining a quality value for a target software based on performance of the target software; and

computing a merit-based licensing fee for the target software based on the quality value.

2. (Original) The method of claim 1, wherein determining a quality value for a target software based on the target software's performance comprises comparing the performance of the target software to performance of another software.

3. (Original) The method of claim 2, wherein determining a quality value for the target software comprises dividing an error rate associated with the other software by an error rate associated with the target software.

4. (Original) The method of claim 2, wherein the other software comprises a free engine.

5. (Original) The method of claim 1 further comprising logging the performance of the target software.

6. (Original) The method of claim 5, wherein logging the performance of the target software comprises evaluating the target software's performance based on field data.

7. (Original) The method of claim 5, wherein logging the performance of the target software comprises evaluating the target software's performance based on test data.

8. (Original) The method of claim 1, wherein determining a quality value for a target software based on the target software's performance comprises computing an absolute value for the quality value.

9. (Original) The method of claim 8, wherein computing the absolute value comprises dividing a predetermined error rate threshold by an error rate associated with the target software.

10. (Original) The method of claim 1, wherein the target software comprises intelligent software.

11. (Original) The method of claim 1, wherein determining the quality value comprises determining a quality value based on a factor selected from the group consisting of accuracy level, efficiency level, throughput level, multilingual capability and self-reporting of errors capability of the target software.

12. (Original) The method of claim 1, wherein determining the quality value comprises computing a value based on at least one point measurement over a timeframe that is substantially equal to or less than a predetermined time period.

13. (Original) The method of claim 1, wherein computing the merit-based licensing fee comprises adjusting a base licensing fee based on the quality value.

14. (Original) The method of claim 13, wherein computing the merit-based licensing fee comprises multiplying the quality value by a first constant and adding the result to a second constant.

15. (Original) The method of claim 1, wherein computing the merit-based licensing fee comprises a floating final licensing cost.

16. (Previously presented) A system, comprising:

a CPU;

a storage device coupled to the CPU and containing executable code; and

wherein, upon executing the code, the CPU computes a merit-based licensing fee for a target software based on a quality value associated with the target software.

17. (Original) A system as recited in claim 16, further comprising an operation log and wherein the CPU logs performance of the target software in the operation log.

18. (Original) A system as recited in claim 16, wherein the CPU compares the performance of the target software to another software.

19. (Original) A system as recited in claim 18, wherein the other software comprises a free engine.

20. (Original) A system as recited in claim 18, wherein the storage device containing executable code further comprises an algorithm that causes the CPU to determine the quality value for the target software by dividing an error rate associated with the other software by an error rate associated with the target software.

21. (Original) A system as recited in claim 16, wherein the storage device containing executable code further comprises an algorithm that causes the CPU to determine the merit-based licensing fee by adjusting a base licensing fee based on the quality value.

22. (Original) A storage device containing software that, when executed by a processor, causes the processor to:

store operation logs of a target software;

measure a performance level of the target software based on the operation logs;

determine a quality value for the target software based on the performance level; and

compute a licensing fee for the target software based on the quality value.

23. (Original) A storage device as recited in claim 22, wherein determining the quality value comprises comparing the performance of the target software to another software.

24. (Original) A storage device as recited in claim 22, wherein computing the licensing fee comprises adjusting a base licensing fee based on the quality value.

25. (Original) A system, comprising:

means for measuring a performance level of a target software;

means for determining a quality value for the target software based on the performance level; and

means for computing a licensing fee for the target software based on the quality value.

26. (Original) A system as recited in claim 25, wherein determining the quality value comprises comparing performance of the target software to another software.

27. (Original) A system as recited in claim 25, wherein computing the licensing fee comprises adjusting a base licensing fee based on the quality value.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.